Dr. Andreas Borchert Institut für Angewandte Informationsverarbeitung 21. Dezember 2009
Wolfgang Kaifler / Michael Mattes

Systemnahe Software I
im Wintersemester 2009/2010

Abgabetermin: bis 19. Januar 2010

Tutorien wieder ab 11.01.2010

Die Tutorien finden wieder ab Montag, 11. Januar 2010 statt.

Beachten Sie auch, dass dieses Blatt erst eine Woche später am 18.01. bzw. 19.01.2010 vorgestellt werden soll.

27 Strings sortieren in Dateien (16 Punkte)

Schreiben Sie ein Programm `linesort.c`, welches zwei Dateinamen auf der Kommandozeile entgegennimmt. Hierbei ist die erste Datei als Eingabedatei vorgesehen. Die zweite Datei soll angelegt bzw. überschrieben werden und die alphabetisch sortierten Zeilen der Eingabedatei enthalten.

Verwenden Sie zur Implementierung der Ein- und Ausgabe die Funktionen `open`, `read`, `write` und `close`. Deren Manpages (*man -s2 open*) könnten hierbei hilfreich sein. Achten Sie auch auf eine korrekte Fehlerbehandlung einschließlich Ausgabe von Fehlermeldungen.

Die Funktionen `fopen`, `fscanf` etc. sollen nicht verwendet werden.

Hinweis: Die Funktion `perror` kann verwendet werden, um eine verständliche Erklärung für den abstrakten Fehlercode der Variablen `errno` auszugeben.

28 Texteditor selbst gemacht (32 Punkte)

Als nächstes sollen Sie einen kleinen interaktiven Editor `myedit.c` selbst schreiben. `myedit` soll beim Start entweder kein Argument (zum Anlegen einer neuen, bisher unbenannten Datei) oder ein Argument (den Namen einer zu öffenden Datei) erhalten.

Damit sich der Aufwand zur Implementierung in Grenzen hält, sind die möglichen Operationen des Editors stark eingeschränkt. Neben `open` zum (Neu-)Öffnen einer Datei innerhalb des Programms gibt es noch die Operationen `print` (Ausgabe des aktuellen Dateipuffers), `change` (Ändern eines einzelnen Zeichens), `replace` (Ersetzen aller Vorkommnisse eines Zeichens durch ein anderes), `insert` (Einfügen eines neuen Zeichens am Ende), `write` ((Zurück-)Schreiben des des Dateipuffers in das Dateisystem) und `quit` (Beenden des Programms).

Bauen Sie auch Rückfragen an passender Stelle in Ihr Programm ein. Beispielsweise muss der Editor beim Speichern den Dateinamen erfragen, falls dieser nicht beim Aufruf über die Kommandozeile oder interaktiv bei `open` angegeben wurde. Ebenso darf das Programm nicht ohne Rückfrage Änderungen verwerfen, wie es z.B. beim Beenden oder beim Öffnen einer Datei geschehen könnte.

Ein Beispielaufruf könnte wie folgt aussehen:

```
$ echo "Hallo" >testdatei
$ ./myedit testdatei
Datei testdatei geladen.
print
Inhalt des Puffers testdatei (nicht veraendert):
Hallo
change 1 u
OK.
change 4 a
OK.
replace l r
OK.
print
Inhalt des Puffers testdatei (veraendert):
Hurra
quit
Puffer enthaelt ungespeicherte Daten. Speichern? (j/n/a): a
Beenden abgebrochen.
open neu.txt
Puffer enthaelt ungespeicherte Daten. Speichern? (j/n/a): n
Datei neu.txt existiert nicht. Lege leeren Puffer an.
insert T
OK.
insert e
OK.
insert s
OK.
insert t
OK.
write
Datei neu.txt gespeichert (4 Byte).
quit
Beende.
$ cat neu.txt
Test$ # Newline am Ende der Datei fehlt
```

Viel Erfolg und erholsame Feiertage!